IJITKM Special Issue (ICFTEM-2014) May 2014 pp. 119-122 (ISSN 0973-4414)

A Survey on Resource Management in Distributed System Using Ant Colony Optimization

Ankush Goel[1] , Sandeep Kumar[2]
[1]Research Scholar Department Of CSE, GITM, Bilaspur
[1,2]Ganpati Institute of Technology and Management, Bilaspur, Yamuna Nagar
[1]ankush.goel120@gmail.com

Abstract: This paper provides an overview of Resource Scheduling in Distributed System. Grid computing is an important and developing computing initiative that involves the aggregation of network connected computers to form a large-scale, distributed system for coordinated problem solving and resource scheduling. The primary concern associated with the grid computing is efficient management of resources, which is a challenging assignment due to distributed and heterogeneous nature of the resources. Resource scheduling is to select the appropriate resource for a particular task in such a way that overall efficiency can be achieved. This paper presents a review of resource management in distributed system. In this paper some of the popular heuristic resources management algorithms are discussed.

Keywords: Grid Computing, Resource Management, Load Balancing, Ant Colony Algorithm, Random Algorithm.

1. Introduction

Grid computing is an important and developing computing initiative that involves the aggregation of network connected computers to form a large-scale, distributed system for coordinated problem solving and resource sharing [1] ,[2]. The main objective of the grid computing is to maximize the utilization of the organization's computing resources by making them as shareable entities, and provide computing on demand to the users. Balancing the load of all available resources is also an important issue in the grid [3]. Scheduling problem in grid system is much more complex as new features of grid systems such as its dynamic nature and the high degree of heterogeneity of jobs and resources must be undertaken [4]. Scheduling is mainly of two types, static and dynamic scheduling. Static scheduling allocates the task to suitable resource before starting the execution. In case of static scheduler all the details of tasks should be known well before the process starts. Dynamic scheduling allocates the resource during the execution time i.e. the scheduler can take decision during job execution [5].

Scheduling can aim to provide two objectives i.e. high performance computing and high throughput computing. High performance computing decreases the execution time whereas high throughput computing increases the processing capacity of the system.

Scheduler has three main phases [6]. In the first phase all the available resource will be collected. This is known as resource discovery. In the second phase collects the information about the resources and choose the best suitable resource to the task. Third phase executes the job in the selected machine.

By distributing the grid resources efficiently, an advanced resource allocation strategy can reduce total run time and total expenses greatly and bring an optimal performance [7].

Heuristic approaches help to solve such type of problems effectively. Opportunistic Load Balancing (OLB), Min-Min, Fast Greedy, Tabu-Search, Max-Min, Minimum Execution Time (MET) and Minimum Completion Time (MCT) Ant Colony Optimization are some of the heuristic approaches which help to solve the grid scheduling problem. The proposed algorithm is based on Ant Colony Optimization (ACO) algorithm which uses batch mode heuristic mapping. In this ant colony algorithm each job is considered as an ant and optimal solution is provided with the help of pheromone detail. Heuristic based ant colony algorithm is used in the second phase of the scheduler.

The organization of the paper further is as follows. The motivation works are discussed in Section II, Heuristic algorithms used for resource management is analyzed in Section III, Section IV describes the problem statement, section V deals conclusion.

2. MOTIVATION

In the past last decade has seen a substantial change in the way we perceive and use computing resources. And also the computational capability and network performance has gone to a great extent but the computing networks are still not fully capable to deal with current complex and resource intensive problems [8]. With the evolution of grid computing many issues has been raised viz. designing, building, deploying and scheduling of various heterogeneous resources in grid environment [9].

To achieve the promising potentials of distributed resources in the grid environment effective and efficient resource scheduling and load balancing algorithms are fundamentally important. Unfortunately scheduling algorithms in traditional parallel and distributed systems, which usually run on homogeneous and dedicated resources can't work well in grid environment.

The motivation of the paper can be summarized as:

1. To aggregate the power of widely distributed resources and provide the non-trivial services to users.
2. Proper utilization of the CPU speed greatly increased in the recent years.
3. Use of the largely unused computing resources.
4. To utilize widespread availability of fast, universal network connection.
5. To improve the efficiency and usability of networks in grid computing environment with high Communication demands.
6. To spread the load equally among the available resources, so as to reduce the waiting time of the tasks and increase the throughput of the resources.

3. HEURISTIC ALGORITHMS USED FOR RESOURCE MANAGEMENT

A lot of research had already been done in the field of distributed environment related to load balancing and found that scheduling problems are NP-hard in nature, which can be better solved by the heuristic technique. Many heuristic algorithms have been designed for scheduling of task in grid environment. But all these algorithms have some drawbacks and still there is a scope to optimize the scheduling process, so as to maximum utilize the resources and increasing the throughput of the resources and the waiting time of the tasks.

Some of the popular heuristic resources scheduling algorithms are [10]:

3.1 Opportunistic load balancing (OLB):

Without considering the job's execution time, it assigns a job to the earliest free machine. If more than one machine is free then it assigns the job in arbitrary order to the processor. This scheduling mechanism runs faster. The advantage of this method is that it keeps almost all machines busy. Yet it does not assure load balance**.**

3.2 Minimum Execution time (MET):

The minimum execution time or MET assigns each job to the machine that has the minimum expected execution time. It does not consider the availability of the machine and the current load of the machine. The resources in grid system have different computing power.

Allocating all the smallest tasks to the same fastest resource redundantly creates an imbalance condition among machines. Hence solution is static.

3.3 Minimum Completion Time (MCT):

The algorithm calculates the completion time for a job on all machines by adding the machine's availability time and the expected execution time of the job on the machine. The machine with the minimum completion time for the job is selected. The MCT considers only one job at a time. This causes that particular machine may have the best expected execution time for any other job. The drawback of MCT is that it takes long time to calculate minimum completion time for a job.

3.4 Max-min

Max-min begins with a set of all unmapped tasks. The completion time for each job on each machine is calculated. The machine that has the minimum completion time for each job is selected. From the set, the algorithm maps the job with the overall maximum completion time to the machine. Again the above process is repeated with the remaining unmapped tasks. Similar to Min-min, Max min also considers all unmapped tasks at a time.

3.5 Min-Min

Min-min algorithm starts with a set of all unmapped tasks. The completion time for each job on each machine is calculated. The machine that has the minimum completion time for each job is selected. Then the job with the overall minimum completion time is selected and mapped to the machine. Again, this process is repeated with the remaining unmapped tasks. Compared to MCT, Min-Min considers all unmapped tasks at a time. The drawback of Min-Min is that, too many jobs are assigned to a single node .This leads to overloading and response time of the job is not assured.

3.6 Ant Colony Optimization Algorithm

Ant colony optimization (ACO)[11] was first introduced by Marco Dorigo as his Ph.D. thesis and was used to solve the TSP problem .ACO was inspired by ants behavior in finding the shortest path between their nests to food source. Many varieties of ants deposit a chemical pheromone trail as they move about their environment, and they are also able to detect and follow pheromone trails that they may encounter. With time, as the amount of pheromone in the shortest path between the nest and food source increases, the number of ants attracted to the shortest path also increases. This cycle continues until most of the ants choose the shortest path. Various types

of ACO algorithm are presented. Each of them has some special properties, i.e., Ant Colony System (ACS), Max-Min Ant System (MMAS). Fast Ant System (FANTS).
In paper [12], they have proposed a simple grid simulation architecture using ACO. They have used response time and average utilization of resources as the evaluation index. In the paper [13] and [14], they have proposed ACO algorithms, which could improve the performance like job finishing ratio. In paper [15], the job is moved from one machine to another machine, so that the traffic in the grid system will be automatically increased. In paper [16], six different ant agents are used. To solve the grid scheduling problem, ACO is one of the best algorithms.

4. PROBLEM STATEMENT

Scheduling a task to a particular resource in such a way so as to improve the execution time and cost is the foremost problem in the grid computing environment. Heterogeneous and dynamic nature of the resources makes it a challenging assignment. For mapping a task to a resource there is a grid scheduler that receives the applications from the grid users, selects the feasible resources for the applications according to acquired information from the grid information service module and submits the resource to the selected resource. The grid scheduler does not have control over the resources and also on the submitted jobs. Any machine can execute the any job, but the execution time differs. As compared with the expected execution time, the actual time may be varied at the time of running the jobs to allocated resource. The grid scheduler's aim is to allocate the jobs to the available nodes. The best match must be found from the list of available jobs to the list of available resources. The selection is based on the predictions of the computing power of the resource. The grid scheduler must allocate the job to the resources efficiently. The efficiency depends on two parameters one is execution time of the job and second is cost of using the resource. To minimize these parameters the workload has to be evenly distributed over all nodes which are based on their processing capabilities. This arises the new issue called load balancing. The main objective of a load balancing consists primarily to optimize the average response time of applications by equal distribution of the load among available resources.
So we can the state the problem as "Map the best resource to a task so that overall execution time and cost of the resource can be minimized."

5. CONCLUSION AND FUTURE WORK

In this research we have consider the above challenges and summarized the goal of work for the dissertation which are as follows:

- We will try to develop few schemes/algorithms which will be adaptive in nature for improving the scalability of the distributed systems. A comparison will also be made with some existing system.
- A new adaptive and efficient method for resource allocation in distributed system will be proposed.
- Focus will be on countermeasures and design considerations for fault tolerant and effective cost & time resource allocation schemes.
- We will study the effect of load migration in P2P networks, Clusters and Grid networks for distributed environment.
- An application prototype will be developed using above adaptive, fault tolerant and dynamic resource allocation methodologies.

References:

[1] Heba Kurdi, Maozhen Li, and Hamed Al-Raweshidy, A Classification of Emerging and Traditional Grid Systems, IEEE Distributed Systems Online, vol. 9, no. 3, 2008, art. no. 0803-3001.
[2] I.Foster, C. Kesselman, and S. Tuecke, The anatomy of the grid, enabling scalable virtual organization, International Journal on Supercomputer Applications, vol. 15, no. 3, 2001.
[3] Salehi, M.A., Deldari, H., Dorri, B.M., 2008. "Balancing Load in a Computational Grid Applying Adaptive, Intelligent Colonies of Ants", Informatics, Vol. 32, pp.327-335.
[4] Foster, I., Kesselman, C., Tuecke, S., 2001. "The Anatomy of the Grid: Enabling Scalable Virtual Organizations", International Journal of High Performance Computing Applications, Vol. 15(3), pp. 200-222.
[5] Chtepen, M., 2005. "Dynamic Scheduling in grid systems", Sixth Firw. PhD Symposium.Faculty of Engineering, Ghent University, 110, pp.1-2.
[6] Schopf, J.M., 2002. "A General Architecture for Scheduling on the Grid", Special issue of JPDC on Grid Computing.
[7] Krauter, K., Buyya, R., Maheswaran, M. (2002), "A Taxonomy and survey of Grid Resource Management Systems for Distributed Computing", Software: Practice and Experience (SPE) Journal, Wiley Press, USA, Vol.32 (2), pp. 135-164.
[8] Kuppani Sathish, A Rama Mohan Reddy, "ENHANCED ANT ALGORITHM BASED LOAD BALANCED TASK SCHEDULING IN GRID COMPUTING." IJCSNS VOL.8 No.10, October 2008.

[9] David De Roure, Mark A. Baker, Nicholas R. Jennings and Nigel R. Shadbolt, Department of Electronics and Computer Science, University of Southampton, Southampton SO17 1BJ, UK " The Evolution of the Grid".
[10] Kousalya.K and Balasubramanie.P,"An Enhanced Ant Algorithm for Grid Scheduling Problem" IJCSNS VOL.8 No.4, April 2008.
[11] M. Dorigo and T. Stützle, Ant colony optimization, Cambridge, Massachusetts, London, England: MIT Press, 2004.
[12] Z. Xu, X. Hou and J. Sun, "Ant Algorithm-Based Task Scheduling in Grid Computing", Electrical and Computer Engineering, IEEE CCECE 2003, Canadian Conference, 2003.
[13] E. Lu, Z. Xu and J. Sun, "An Extendable Grid Simulation Environment Based on GridSim", Second International Workshop, GCC 2003, volume LNCS 3032, pages 205–208, 2004.
[14] H. Yan, X. Shen, X. Li and M. Wu, "An Improved Ant Algorithm for Job Scheduling in Grid Computing", In Proceedings of the Fourth International Conference on Machine Learning and Cybernetics, 18-21 August 2005.
[15] Li Liu, Yi Yang, Lian Li and Wanbin Shi, " Using Ant Optimization for super scheduling in Computational Grid, IEEE proceedings of the 2006 IEEE Asia-pacific Conference on Services Computing (APSCC' 06).
[16] R. Armstrong, D. Hensgen, and T. Kidd, "The relative performance of various mapping algorithms is independent of sizable variances in run-time predictions," in 7th IEEE Heterogeneous Computing Workshop, pp. 79–87, Mar. 1998.